

Title	推論エンジン, 「論語」論理の場合(計算理論とその応用)
Author(s)	高橋, 英之
Citation	数理解析研究所講究録 (1997), 992: 159-166
Issue Date	1997-05
URL	http://hdl.handle.net/2433/61142
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

推論エンジン、「論語」論理の場合

Deductive Engine for the Logic of the Analects of Confucius

日大理工数 高橋英之 (Hideyuki Takahashi) *

Abstract

I made a deductive engine for simulating some aspects of the basic reasoning seen in the Analects of Confucius (AoC). This paper reports its design philosophy. The logic of AoC is that of "value", and the value is mainly brought from the planning for aiming the ultimate goal of "the happiness of people". Our engine is basically similar to a planner; it makes a branching tree led by "hot proposition" and can be of the confluent type. We treat also "emotion" such as delight and sorrow. It is assumed the emotions are immediate results of value; positive value causes positive emotion, i.e., delight, and negative value causes negative emotion, i.e., sorrow. The engine has the value calculation system, which causes emotion.

We can say that the deductive engine, or AI-Confucius system has begun moving, though the move is only a toddle.

1 要約

自然言語の意味理解には、プランニングが重要であることが認識されている。プランニングは行動の目的と方法に関係し「価値」の問題と関わりが深い。ロボットでも自由行動を始めた暁には価値の問題に遭遇する。真偽を問題とする普通の論理と異なって、価値論理は善悪を問題にする。筆者は特に「論語」における価値論理を研究しているが、最近この目的にかなった推論エンジンの原型を Prolog で作成して単純な推論ができるようになったので、本講演ではそれについて報告したい。

推論エンジンは概ねプラナー (planner) の一種である。一種の定理証明機でもあり「論語」論理の推論のインタプリタである。我々の推論エンジンは大別して (1) 全命題型、(2a) ホット命題型で分岐型、(2b) ホット命題型で合流型、の3種に分けることができる。1と2bとは等価であり、2aは不十分である。また別の、通常分類である技法から見た分け方では、パス (path) 型の探索と木型の探索があるし、縦型と横型、前向き推論と後ろ向き推論があり、それぞれ実現した。状態を構成する命題の記述には格 (case) 形式を使う。但し単一化のときの付帯条件が可能な形とした。もう一つのポイントは目標状態をどう取るかがであり、それには特定状態を目標とする通常のものの他に、dead-end 状態をとることも可能とした。

「喜び」「悲しみ」等の感情を扱う。感情は価値に比例する、特に、両者のプラス・マイナスの符号が一致する、と仮定する。そのとき問題は、「価値の計算システム」を作ることになる。これは基本的なものの価値を設定し、他は基本からの導き方を規定することと、+・- 掛け算その他から算定できる。これは筆者の提案である。

こうした概して既知の技術から遠くは離れていない方法で「論語」冒頭の章を扱う。この「論語」論理推論エンジンについて今の段階で言えることは、まだたどたどしいけれどもたしかに動き始めている。まだ全く不十分ではあるが、出発点は設定できた、ということである。

* E-Mail: tkh@math.cst.nihon-u.ac.jp

分野：人工知能、自然言語理解

キーワード：価値論理、自動推論、プランニングとプラナー、自然言語の意味論・文脈論、「論語」論理

2 導入：状態と CD、行為は状態変化オペレータ

論理とプランニングは類似点がある。論理においては、(1) A, (2) A—推論規則—>B, (3) therefore, B という推論が基本であるが、プランニングにおいては、(1) 太郎は東京に居る (状態 S1), (2) 太郎は東京から京都へ行く (行為 A), (3) 太郎は京都に居る (状態 S2), というふうに、(1) 状態 S1, (2) 行為 move(S1,A,S2,C), (3) 故に、状態 S2, と推論する。つまり論理の定理にあたるのは状態 (命題集合) であり、推論規則にあたるのは行為である。但し C は確実度である (本当はファジー論理を扱うべきだが、本稿では立ち入らない)。これを形式化すると、

(1) [居る, actor: 太郎, at: 東京]	状態 S1
(2) move([prestate: [居る, actor: X, at: P]],	前状態
[移動する, actor: X, from: P, to: Q],	行為の一般化記述 A
[poststate: [居る, actor: X, at: Q]],	後状態
[確実]).	確実度
(2') [移動する, actor: 太郎, from: 東京, to: 京都],	行為実例 A'
(2'') match(A,A')	A と A' をマッチ
(3) [居る, actor: 太郎, at: 京都]	状態 S2

となる。行為とは状態に作用するオペレータである。一般には、行為前後で '不変' な命題 condition: proposition(s) を付ける。ここで用いた命題の意味表記法、[居る, actor: 太郎, at: 東京] 等は、格 (case) 形式と呼ばれる [Fillmore68, Ikegami75]。Schank らはそれを CD (Conceptual Dependency) と呼んだ [Schank81]。格とは行為記述に関わる概念だが、しかしデータベースの基本形式は、

名前 (name)、および、キーワード (keyword) と値 (value) の組の集まり

である。上記の命題形式もそれに似た、表題=動作名、キーワードと値の組の集まり、から成っている。ちょうど市役所の書類が、表題=住民登録、住所: どこそこ、氏名: だれそれ、といった形式であるのと類似している。だから、この形式を特に格形式と呼ぶ必要は必ずしもない。データベースの一形式だと考えればよい。その意味で、我々はこの形式で書かれたものを、"コンセプト・データ (概念データ、Concept Data, CD)" と呼ぶことにしたい。要するにそれは小さなデータベースであり、文章の "意味" とは、小さなデータベースである、と考えるのである。

我々は上記のような CD に、マッチのときの付帯条件のついた形、例えば

([移動する, actor:X, from:P, to:Q] :- not identical(P,Q))

を許す。この付帯条件は後続の過程でもずっと有効であるとする。

上では「移動する」を "素過程" と考えた。これをもっと詳細に分解することもできるし、移動の方法を多く並べることもできる。何を素過程ととるかは設定の自由である。

3 「論語」一の 1、学・教・政

「論語」冒頭の章 (学而第一の 1) は、小論語とも論語のエッセンスのエッセンスとも言われるものだが、この章は、「子曰く、(1) 学びて時にこれを習う。また悦 (悦) ばしからずや。(2) 朋遠方より来たる有り。また楽しからずや。(3) 人知らずして 慊 みず。また君子ならずや。」という 3 節から成っている。問題は 2 点あり、1 つは表層から深層へ、即ちこの自然言語表現を、意味の形

式的表現に機械的に変換することであるが、本稿はこの問題には関わらない。2つめは深層レベルにおいて、その意味を、なんらかの論理形式のうちに位置付け、コンピュータに演繹をさせて、なるべく少ないものから、できるだけ多くのものを証明させることである。一種の文脈論であり、本稿はこれにのみ関わる。/「論語」には沢山の注釈書があり、上述の章についても色々な解釈がある[kanno15]ほか。まず、この文の主語は誰か、という点である。

(a) 主語は孔子である。するとこの章は孔子の一生を表すことになり、上記(1)は「学習」、(2)は「教育」、(3)は政治的地位への就職、そこでの礼治「政治」、そしてそれによって「人々の幸福」を目指す、但し孔子は就職に失敗した、という意味となる。

(b) 主語は弟子である。すると、これは孔子の学校での校則ともなり、(1)は学習、(2)は学友との交流(学習の一環)、(3)は先と同じく政治的地位への就職云々、となる。

なお、ここの「学習」は「礼」の習得という限定された意味であり、礼とは制度・慣習法・儀礼で、孔子の学校は今で言えば、法学部の学生に法律を教えながら、人格も大事だと説教しているようなものである。色々な解釈は相互に排他的なものではなく、この章はそれらの解釈をすべて含んでいるのだ、と考えてよからう。おおよそこうした解釈がすべて導出されるならば理想的である。推論エンジンを作り、推論の前提(公理に当たるもの)を作る作業は、「論語」という”現象”に対する”本質”を探究する営みであり、自然科学の方法と似た点がある。まず行為の原型として以下のmove/4を設定する。これが骨格となる。内容は学習、就職、政治、教育の4つの行為である。

```
move( [prestate: [ not('獲得した'), subject: X, object: '礼' ]],
      % つまり 「Xが礼を 学んでいない 状態」から、
      [ '****学****', subject: X, object: '礼' ],
      % つまり 「Xが礼を 学ぶ という行為」によって、
      [poststate: [ '獲得した', subject: X, object: '礼' ]],
      % つまり 「Xが礼を 学んだ 状態」に至る。
      [reliability: '確実'] ). % <== 確実さ、の程度。
move( [condition: [ '獲得した', subject: X, object: '礼' ]],
      % つまり 「Xが礼を 学んだ 状態」から、
      [ '****就ける****', subject: '為政者'(Y),
        object: X, to: '政治的地位'(Z) ],
      % つまり 「為政者YがXを 政治的地位Zに 就ける 事によって、
      [poststate: [ '居る', subject: X, at: '政治的地位'(Z), range: '国' ]],
      % つまり 「Xが 政治的地位Zに 就いた 状態」に至る。
      [reliability: '概然的'] ).
move( [condition: [ '居る', subject: X, at: '政治的地位'(Z), range: '国' ]],
      % つまり 「Xが 政治的地位Zに 就いた 状態」において、
      [ '****礼治政治****', subject: X, to: ('人々'), range: '国' ],
      % つまり 「Xが 国の人々に対して 礼治政治をする 行為」によって、
      [poststate: [ 'である', subject: ('人々'), range: '国',
        complement: '幸福' ]],
      % つまり 「国の人々が 幸福 である 状態」に至る。
      [reliability: '確実', base: '実績'] ).
move( [condition: [ '獲得した', subject: X, object: '礼' ],
      prestate: [ not('獲得した'), subject: '学生'(Y,to(X)), object: '礼' ]],
      % つまり 「Xが礼を 学んだ 状態」から、
```

```
[ '****教****', subject: X, object: '礼', to: '学生'(Y,to(X)) ],
  % つまり 「Xが 学生Yに 礼を 教える という行為」によって、
[poststate: [ '獲得した', subject: '学生'(Y,to(X)), object: '礼' ]],
  % つまり 「学生Yが 礼を 学んだ 状態」に至る。
[reliability:'确实'] ).
initial_state( [[ not('獲得した'), subject: '孔子', object: '礼' ]]).
initial_fact( [ not('獲得した'), subject: X, object: '礼' ] ):- X \= '孔子'.
final_state( [[ 'である', subject: ('人々'), range:'国',complement: '幸福']] ).
```

これで initial_state(Si) で与える初期状態 (命題集合) Si から出発して initial_fact(Fi) で与える命題 Fi をも利用しながら、final_state(Sf) で与える目標状態 Sf を達成するような行為系列 A1・A2・...・An を求めることが、今の課題である。

我々の研究は、「論語」を”現象”としてそれを導出できる”本質”を見出すことであるが、現象から本質へは飛躍があるのは通常自然科学と同じである。上記の move/4 は「行為とは状態変換オペレータである」との見方に基づいており、行為の部分は「論語」に見受けられるが、状態の部分は「論語」に必ずしも見受けられない述語 (例えば”幸福”) を用いている。

4 全命題型のプラナ、「教」の無限系列

プラナには各種の分類がある。まず取り上げたいのは、原初的なプラナである全命題型のプラナである [Sterling94]。これは行為により新しく付加された命題を他と区別せず、袋に入れてしまうタイプで、後述のホット命題型と対照される。[全命題, そのパス] を状態として、縦型 (深さ優先)・パス (path) 型・前向き (ないし後ろ向き) の探索を行なうものであり、Prolog で書かれている。プランの出力は、状態::=>行為==>状態::=>状態==>行為... の形とする。さて、このタイプのプラナによって多数のプランが得られる。それらの中には、当然予想される、学=>就職=>礼治政治=>人々の幸福という形のものが入っている。

```
[[not 獲得した,subject:孔子,object:礼]]
::=> [****学****,subject:孔子,object:礼]
==> [[獲得した,subject:孔子,object:礼]]
::=> [****就ける****,subject:為政者(X),object:孔子,to:政治的地位(Y)]
==> [[獲得した,subject:孔子,object:礼],[居る,subject:孔子,at:政治的地位(Y),range:国]]
::=> [****礼治政治****,subject:孔子,to:人々,range:国]
==> [[獲得した,subject:孔子,object:礼],[居る,subject:孔子,at:政治的地位(Y),range:国],[である,subject:人々,range:国,complement:幸福]]
```

しかし、更に、以下のようなプラン、即ち、自分の学=>弟子を教育=>弟子が就職=>弟子が政治=>人々の幸福という系列もあり、これは孔子がその弟子を通して目標を達成するプランであると理解できる。途中を示す。

```
==> [[獲得した,subject:孔子,object:礼]]
::=> [****教****,subject:孔子,object:礼,to:学生(X,to(孔子)))]
==> [[獲得した,subject:孔子,object:礼],[獲得した,subject:学生(X,to(孔子)),object:礼]]
```

::==> [****就ける****,subject:為政者(Y),object:学生(X,to(孔子)),to:政治的地位(Z)]
以下略。以下の主人公は、学生(X,to(孔子))である。

さらには、孫弟子'学生'(Y,to('学生'(X,to('孔子')))))そして、曾孫弟子、...、の系列が、次々に得られる。これは、ほんのちょっとした工夫、即ち、「教」の中の表現'学生'(Y,to(X))によって可能となった。但し、Yは学生の名前、Xは先生である。この先生の所に、また再帰的に、学生が来てよい。こうして、以上のような move/4 の設定により、孔子自身の就職・政治を導出できるのと同時に、弟子達のそれ、即ち弟子を通じての目的達成をも、併せて導出できることになる。

この全命題型のプラナでは、自分の学習、そして様々な世代の、教育、就職、政治、が、部分的な順序を除いては”順不同”で現われる。これはこれで良いのだが、次にこの点を改変することを考える。

5 ホット命題型のプラナ、有限オートマトンと行為の正則表現

形式的推論における証明を見れば明らかなように、今ちょうど証明された”ホットな命題”に対して次段の推論が施される。この考え方をプラナに当てはめると、ある時点で適用する move/4 に対して、以下の制限を加えることになる。すなわち、各時点での状態とは、前節では[全命題, そのパス]であったが、それを状態 = [ホットな命題, そのパス, 全命題, そのパス] という4つ組とする。そして move(S1, Action, S2, Rel) が適用され得るのは、S1 とホットな命題とが交わりを持つ場合のみである、とする。このタイプのプラナを作成して実行すると、就職に関して、本人、弟子、弟子の弟子、...、の系列が順序正しく、直接の因果関係を示しつつ現われる。つまり、

学習、教育ⁿ(各弟子に対する)、就職(弟子ⁿの)、礼治政治(同)

といった行為系列(プラン)が得られる。ところでこれらの集まりは、ある簡約化を行なえば、正則表現として表しうる。即ち、学習・教育・就職・礼治政治である。簡約化しなければ、弟子ⁿが現われる以上、有限のものの繰り返しではあり得ないけれども。

正則表現が出てきたので、あらためて考えれば move(S1, Action, S2) の集合は、状態 S に対するいくつかの簡約化を行なうなら、有限オートマトンと見なすことができ、move は状態遷移を定義している。するとプランニングとは、ある簡約化のもとで、有限オートマトンからその受理集合を正則表現として求める、という問題となる。筆者はこの観点からもいくつかのソフトを作成した。

6 ホット命題・木型・合流型のプラナ

各時点でのどの行為を行なうかの場合分けは、いわゆる探索木であり、パス型では探索木は観念としてのみあった。しかしプラナが実際に、探索木をデータとして持って段々作成していくというタイプが可能である。筆者は [Bratko90] が書いているプログラムを原型にして、まず「ホット命題型で木型タイプのプラナ」を作成した。

プラナが持つ木は、探索木だと考えることができる一方、ホット命題型の場合には、アナロジーとしては証明図になぞらえることができる。ところで、Gentzen の自然推論の証明は木状 (tree-form) ではあるが、正確にはそれは「合流型」つまり「合流をも併せ持つ木」である。例えば AND-導入がそれである。プランニングにおいて実際に、行為前条件が複数命題であるような行為として、前出の「教育」の他に、2人の学生の交流、がある。つまり、先生のもとへ独立に2人の学生 X と Y がやって来て、そこで2人が交流する、というものである。これは合流を使わない定式化も可能であろうが、合流によって自然に取り扱うことができる。詳細は略。このような行為を扱えるものとし

て、ホット命題型の「合流型プラナ」を Prolog で作成した。これがプラナとして現在筆者が持っている最も一般的なプラナである。

ところで、合流型プラナが「論語」論理で利用できる本質的理由は、木の各パス (path) が、世界で平行して生じ得る、と仮定できることにある。つまり、政治をやりながら、教育もやる、というふうな並列的行為が「論語」では可能だとされていることである。

7 感情と価値：感情は価値に比例する

「論語」冒頭の章では更に、「説（悦）ばしい」、「楽しい」、「^{うれ}む」など、広い意味での感情が述べられている。これを扱うことを考えたい。筆者の研究の現段階では、感情もまた記号的に記述されるのみである。我々はこれらの感情を取り扱うのに、第一近似としては、「感情とは、プラスないしマイナスの値そのものである」として取り扱いたい。符号が重要であり、数値は二の次である。そこでまずは、プラスの感情として「喜ぶ」、マイナスの感情として「悲しむ」という2つだけとする。これらにより微妙な差を与えることは今後の課題である。

これらの感情の値が、実は「価値」に付随したものであることを認識したい。「感情のプラス／マイナスの符号は、価値の符号と一致する」と仮定する。その値もおおむね比例関係にあるものとする。そう仮定すれば「論語」をシミュレーションできる。例えば以下のようにプログラムできる。

```
move([condition:
    ( CD :- once: signestimate2complement(CD, Compl, Subject1, Reason, Rest)),
    % CD は変数。once は唯1回だけ実行されることを意味する。
    % Compl は補語、Subject1 は誰が感情を持つかを求める。
    [ '****(心理の) 因果的推移 ****', subject: Subject1 ],
    [poststate: ['である', subject: Subject1, complement: Compl, 詳細は略。
signestimate2complement(CD, Compl, Subject1, Reason, Rest):-
    sign_of_value( CD, Sign, Reason, State ), % CD の「価値」の符号 Sign を求める。
    ( Sign = + -> Compl = '喜ぶ状態' % CD が良き事であるなら喜ぶ状態になる。
    ; Sign = - -> Compl = '悲しむ状態' % CD が悪き事であるなら悲しむ状態になる。
    ), /* 「善い状態は嬉しい。悪い状態は悲しい」という心理法則。以下、略。
% 以下の2つのうち、いずれか一方を選ぶ。ここを複数にすると「循環」が生じうる。
is_subject('孔子'). %is_subject('私'). % <-- 私 = 弟子
```

最も問題なのは「価値の計算」の部分である。考え方は、まず基本となるものの価値を定め、他は、基本価値からの導出規則によって導き出す、というやり方である。以下のような仮定に基づいて計算する。

(0) まず、究極善は、「人々」および「幸福」である（と仮定する）。

(1) 状態を表す述語である「在る」にはプラス、「無い」にはマイナスの符号を与える。これに近いものとして、「何かに近づく」、「何かを望む」など、正負の価値に対して肯定的（否定的）な述語にはプラス（マイナス）の符号を与える。

(2) 目的-方法原理。目的が善=>方法が善、という原理を仮定する。即ち、目的-方法関係は、価値を継承する。例えば、

礼 --[方法]----> 人々の幸福 により、価値が遡行する。
+価値 <-----+価値

(3) 部分原理。全体が善=>部分が善、という原理を仮定する。即ち、全体-部分関係は、価値を継承する。類似のものに、類-種関係や、集合-成員関係もある。例えば、

人 --[部分]---> 人々 により、価値が遡行する。
+価値 <----- +価値

(4) not(X) の価値は、X の価値にマイナスを掛けたものだ、と仮定する。

(5) +・— 掛け算則。次の規則を仮定する。これにより、「善きもの(+)を望む(+)ことは善(+×+=+)である」といった推論ができるようになる。

```
plus_minus_product( +,+,+ ).   i.e., + × + = +
plus_minus_product( +,-,- ).   i.e., + × - = -
plus_minus_product( -,+,- ).   i.e., - × + = -
plus_minus_product( -,-,+ ).   i.e., - × - = +
```

以上の計算原理によって色々な、物・状態(感情を表す補語を含む)・行為の価値を定めることができる。但し残念ながら筆者はまだ、これらをプログラムはしたが、推論のデータを与えてコンピュータに推論させるという形にはなり切っていない。以下にデータベースの一部を示す。

```
value_of_material('人々',[sign(+),num(100)]).
% '人々' は、究極の価値 (の1つ) である。
value_of_material(X,Value):-
    isa(X,Y), value_of_material(Y,Value).
% 集合-成員関係や、類-種関係は、価値を継承する。
value_of_material(X,Value):-
    partof(X,Y,Coeff), value_of_material(Y,Vy), multiply_coeff(Vy,Coeff, Value).
% 全体-部分関係は、価値を継承する。
value_of_material(X,Value):-
    plan_for(X,Y,Coeff), (is_state(Y) -> value_of_state(Y,Value1) 以下、略。
% それが何かの手段であるなら、その目的の価値を引き継ぐ。
```

注意。['****(心理の) 因果的推移 ****', subject: Subject1] と書いた。これはもはや行為とは言えない。心理法則であり、世界の運行を表す法則である。したがってこれに基づいて得られるのはもはやプランとは言えない。行為と法則を含む、推論と予測の系列である。ソフトは、プランというよりプルバ(prover)に近づく。この事態に合わせて最終状態も変えなければならない。我々ここでは最終状態を、デッドエンド(dead_end)、即ち、もうそれ以上、如何なる推論をも施すことができない、ということをもって、最終状態だとする。

8 plan・do・see 計画・実行・検証

さて、これまではおもに「プランニング」の話であった。では「実行」はどうなるだろうか？プランニングは move/4 の上で行なわれるが、これは”観念”の世界である。一方、実行は”現実”の世界のことである。観念と現実とは、相対応しているだろう、しかし食い違いもありうる。両者は「環境」が異なる、と考える。ここでは、実行の場面で、ある自他の行為が、実際には起こり得なかった、不可能であった、という場合を扱いたい。例えば、

```
impossible(
    [ '****就ける****', subject: '為政者'(X), object: '自分', to: '政治的地位'(Y) ] ).
```


という事実（ここでは自分＝孔子）があった、と仮定して、実行の場面を少し扱いたい。要点は、実行は、プラン・チェックに該当するのではないか、という処にある。証明 (proof) に証明チェック (proof-check) があるように、プランにもプラン・チェックがありうる。(1) まずプランで複数のプラン（第1のプラン、第2のプラン、...）を得る、(2) 次に、それらプランを、現実に対応した、即ち impossible/1 が置かれた環境においてチェックする、(3) 第1のプランに失敗した場合には、第2のプランを実行する、という3段階をとる。これは、plan_do_see（計画・実行・検証）である。つまり我々は、plan_do_see という思考パターンを借りてきて、「論語」に見られる孔子の行動を理解しよう（コンピュータにシミュレートさせよう）というわけである。具体的には先ず (1) プランとして、

(i) 学=>就職=>礼治政治=>人々の幸福（但し現実には、就職が不可能）

(ii) 学=>教=>弟子の就職=>弟子による礼治政治=>人々の幸福

などを得た、とする。次に (2) この (i) の実行は失敗する。挫折である。そこで次に (ii) を実行する、という順序となる。孔子は（大まかに言って）現実政治家としては失敗して、弟子の教育へと志向を変えた。その孔子の行動が、plan_do_see によってシミュレートできる、と言いたい。

9 まとめ

「論語」論理に適合した推論エンジンの原型を Prolog ソフトとして作成した。「論語」論理は、価値論理の1つであるが、これを「目的とプランニング」として把握して、コンピュータ上に実現しようとした。コンピュータによる「論語」論理のシミュレーション、と言ってもよい。感情も、価値に比例するものとして、価値の計算から適切に導くことができた。

まだ、冒頭の章+ α しかできていなくて、しかも、その動きはたどたどしい。しかし確かに、動き始めている。第一歩は踏み出すことができた、と言ってよい。課題は、特に価値計算の所が、プログラム化したけれども、材料を与えてコンピュータに推論させる形になり切っていないこと。そして何よりも、もっと多くの「論語」命題を取り扱いたい、ということである。

我々の中期目標は、「論語」論理に対するデータベースと推論エンジンを完成することであり、長期目標は、孔子との対話システムの作成である。この企ては、なんとしても実現させたい。

参考文献

- [Bratko90] I. Bratko, *PROLOG-Programming for Artificial Intelligence*, Addison-Wesley Pub. Co., 1990. (近代科学社より「Prolog と AI」2分冊で部分翻訳あり).
- [Fillmore68] C. Fillmore, "The case for case", in Bach and Harms(eds), *Universals in Linguistic Theory*, Holt, Rinehart, and Winston, 1968.
- [Ikegami75] 池上嘉彦、「意味論—意味構造の分析と記述」、大修館書店、1975、1993（7版）.
- [kanno15] 簡野道明、「論語解義」、明治書院、1915.
- [Schank81] R. C. Schank, C. K. Riesbeck, *Inside Computer Understanding*, Lawrence Erlbaum Associates, Inc., 1981. 石崎他訳、「自然言語理解入門—LISP で書いた5つの知的プログラム」、星雲社、1986.
- [Sterling94] L. Sterling, E. Shapiro, *The Art of Prolog, 2nd Ed.*, The MIT Press, 1986（初版）, 1994. 松田訳、「Prolog の技芸」、共立出版、1988.